REHOSTING THE AEGIS EMBEDDED TRAINER IN COMMERCIAL PRODUCTS

Mr. Gregory D. Miller and Mr. Jeffrey Lipe

With acquisition reform, the Navy has begun upgrading shipboard systems to adopt commercial off-the-shelf (COTS) technology. The transition of the MIL-SPEC (Mk 29) Aegis combat training system (ACTS) to a COTS system (Mk 50) is an example of such an endeavor that has been successfully achieved. The ACTS rehost effort included changing the system's high-order language, real-time operating system (RTOS), compilers and linkers, development environment, target computing system, and external interfaces. COTS technologies such as VMEbus, local- and wide-area networks, IRIG-B time signals, ethernet, FDDI and SCSI interfaces, and magneto optical disks were all exploited in the upgrade. Definition of a viable architecture and selection of the appropriate hardware, RTOS, and programming language were key to the project's success. Many of the methods used and lessons learned during this effort can be applied to other MIL-SPEC-to-COTS technology rehost projects.

BACKGROUND

ACTS is a core element of the Aegis weapons system (AWS). ACTS is an embedded, hierarchical training system consisting of three major subsystems. The Lesson Control Program (LCP) provides training for basic operator actions at a combat information center (CIC) submode, executing commands from prescripted lessons. The Training Control Program (TCP) provides warfare team, integrated CIC, and battle-force training using prescripted or sailor-modified scenarios. The Postmission Analyzer (PMA) provides debrief capabilities for TCP training exercises using Aegis data recording as input.

ACTS Mk 29 initial implementation was in Aegis Baseline 0, in the fourth bay of the AN/SPY-1A AN/UYK-7 computer suite. When the AWS transitioned to AN/UYK-43 computers in 1986, ACTS migrated to the AN/UYK-43 computer used for "N+1" redundancy. Given this environment, the same basic computing system architecture, computer program architecture, topology, language, operating system, and tool set was used for the first 12 years of ACTS development and maintenance.

In 1991, training improvement requests from the Aegis Training Center (now the Aegis Training and Readiness Center, or ATRC)—such as a scenario modify/save capability and increasing the training track capacity—were unachievable due to the limitations of the AN/UYK-43. The Training and Simulation Branch began investigating options to improve

processing capabilities. Initial recommendations for upgrading ACTS were disapproved by the Aegis Program Office. The training improvements alone did not justify the expense of enhancing the system. In December 1991, the Battle Force Tactical Trainer (BFTT) Program Office presented the BFTT program plan to the Aegis Program Office. The Aegis Program Office subsequently funded General Electric (GE) (now Lockheed Martin) to determine if an ACTS/BFTT training configuration was feasible for Aegis cruisers and destroyers. A study group (including GE, ATRC, the Naval Surface Warfare Center, Dahlgren Division (NSWCDD), and Aegis Program Office representation) recommended a high-level ACTS/BFTT training system architecture and a general plan of action. However, the group did not recommend a specific ACTS system architecture. BFTT requested Aegis participation in BFTT's Demonstration Test (DT) IIA, but due to lack of resource availability, GE could not implement the plan until after the completion of AWS Baseline 6 Phase I. NSWCDD was subsequently given authorization to implement the BFTT DT-IIA requirements in ACTS Mk 29. NSWCDD then used DT-IIA as a proof of concept to justify the conversion of ACTS Mk 29 to a fully COTS environment. The success of DT-IIA resulted in approval for NSWCDD to rehost ACTS for lifetime support engineering (LSE) baselines (6 Phase 1 and below) to provide BFTT/ACTS training capabilities. In addition, the increased processing power of the new system would allow the new ACTS system to provide the ATRC with requested training improvements. This article describes the engineering performed to rehost the ACTS computer programs and successfully deliver the COTS system to the fleet.

TASK

The primary objective of the ACTS Mk 50 effort was integrated ACTS and BFTT training. The main ACTS/BFTT interface was to be the Synthetic Theater of War (STOW) FDDI local area network (LAN), based on IEEE-1278.1 (IEEE Standard for Distributed Interactive Simulation–Application Protocols). The Training Program Development Group weighed various architecture options to integrate ACTS and BFTT and to increase the ACTS track file capacity to 2000 (for BFTT). The group considered alternatives on the spectrum from staying in the UYK-43, to completely redesigning and rehosting ACTS in a symmetric multiprocessing (SMP) environment. However, performance requirements eliminated any solution based on an AN/UYK-43 host; schedule requirements precluded a complete redesign to an SMP environment.

In January 1995, NSWCDD's Training and Simulation Branch of its Combat Systems Department presented a design plan to the Aegis Program Office for a computing system that met the track and interface requirements imposed by integrating with BFTT. The plan was accepted and the Training and Simulation Branch was tasked to rehost ACTS Mk 29 functionality into a COTS-based system that would provide BFTT integration.

The task of rehosting the ACTS computer programs included identifying the development system, high-order language, target computing system components, and RTOS. Approximately 40,000 CMS-2 statements for two programs (TCP and LCP) were involved in the rehost. A third application (PMA), consisting of 30,000 CMS-2 statements and obsolete FORTRAN drivers, was implemented by reusing Aegis data analysis and reduction (ADAR) code. Two teams were established—one for architecture definition and one for component selection—to perform the rehost effort.

A thorough understanding of the current and planned ACTS processing requirements was requisite to defining the components of the new ACTS system. Each message defined for ACTS was sized for maximum sustained and peak throughput requirements. Planned messages for BFTT were estimated based on best engineering data. Various architectures were postulated and sized given the known environment. The defined throughput of the ACTS system is shown in Table 1.

The Training Program Development Group envisioned the new ACTS as a system of upgradable and expandable building blocks. The group emphasized selection of mainstream components that supported multiple, associated COTS products. For example,

Table 1—Defined Throughput of the ACTS System

Interface	Sustained Throughput	Peak Throughput
ACTS/C&D	50,000 bytes/s	84,000 bytes/s
ACTS/SPY	25,000 bytes/s	25,000 bytes/s
ACTS/WCS	2,000 bytes/s	2,000 bytes/s
ACTS/AEGIS Clocks	8,200 bytes/s	8,200 bytes/s
ACTS/RD-358	180,000 bytes/s	180,000 bytes/s
ACTS/Optical Disks	230,000 bytes/s	520,000 bytes/s
ACTS/BFTT Clock	100 bytes/s	100 bytes/s
ACTS/BFTT Simulations	890,000 bytes/s	890,000 bytes/s
ACTS/TAC Host	0 bytes/s	9 Mbytes (load) 10,240 bytes
ACTS/OCD	135 bytes/s	135 bytes/s

any RTOS that supported only one processor was dismissed as not viable because of its self-limiting nature. Additionally, each component was scrutinized for conformance to commercial standards, such as:

- ANSI/IEEE 1014 (VMEbus)
- ANSI/IEEE 1096 (VME side bus (VSB))
- ANSI X3.131 (SCSI)
- IEEE 802.3 (ethernet)
- ANSI X3T9.5/X3.139/X3.148 (FDDI)
- IEEE 1278 series (Distributed Interactive Simulation)
- IEEE 1003 series (POSIX)

However, it was determined that simply being compliant with standards was not enough. Many standards are open to interpretation and provide different levels of compliance. The VMEbus standard, for instance, provides five decreasingly "important" classifications of guidelines (rules, recommendations, suggestions, permissions, and observations) that a manufacturer may or may not implement and still be fully compliant. In the case of the RTOS, each candidate (HP-RT, LynxOS, pSOS, and VxWorks) claimed to be "POSIX compliant." Given the nature of the standard, however, the architecture team was required to define the subset of POSIX standards that were significant to ACTS and select the RTOS based on the level of compliance that was possible without needlessly sacrificing system performance.

Selecting Components

The Training Program Development Group learned quickly during component selection that it was necessary to strike a balance between the conflicting goals of gathering the information needed to make informed decisions and the need for meeting a deadline. There was a major risk of "analysis paralysis" due to the overwhelming volume of choices available, many of which could have adequately served the desired purpose. On the other hand, the group was aware that "ignorance is voluntary misfortune," and labored to avoid the results of overly hasty decisions. In the end, up-front choices were limited by defining the required general characteristics of the components, eliminating nonmatches, and performing more thorough analysis of the fewer remaining candidates. Specific components were selected with an emphasis on open-systems design goals. During selection, each component was considered a part of the overall system, as defined in the following subsections.

Topology Considerations

The ACTS Mk 50 computing system had to support both legacy Naval Tactical Data System (NTDS) interfaces for communication with the AWS and new COTS interfaces for BFTT and intrasystem communication. The requirement for the continuation of six legacy interfaces to the AWS made a

VMEbus-based system—at least to support NTDS input/output (I/O)—inevitable. No other available mainstream solution could satisfy this constraint. Communications with several legacy onboard trainer (electronic warfare and underwater) systems that were previously NTDS-based were to be changed to be accomplished via LAN Access Units (LAUs), which convert NTDS messages to/from LAN messages.

In addition, a new ethernet interface for computer program booting and operator control; an IRIG-B LAN for external ship clock synchronization; and an FDDI LAN for communication to the BFTT, operational readiness and test system, and Aegis display system were requirements for ACTS Mk 50. Given that ACTS is a shipboard system, size and location was a major consideration in the topology definition. The ship integration engineers from the initial ACTS/BFTT integration study group determined that the "N+1" UYK-43 computer should be retained for redundancy, and they further decided that the ACTS Mk 50 would be collocated in one of the new BFTT computer racks. Therefore, even though various options had been considered for hosting the ACTS applications in a physical unit separate from the I/O VME, such as a workstation, space limitations led to the conclusion that the entire system would have to be hosted in the same VME enclosure.

Early Configuration Management Considerations

From the beginning, the Training Program Development Group considered each hardware, firmware, and software component of the ACTS Mk 50 system from a configuration management (CM) aspect. COTS product vendors do not generally treat product upgrades with the level of discipline that safety and mission-critical systems require. Products can be altered at will and without identifiers (such as version numbers). The group observed that vendors of stable, embedded, real-time system components did not require frequent corrective releases of their products. Additionally, the group considered the size of various vendors' operating systems and device drivers. In 1994, it was difficult to imagine CM of a 400-megabyte Unix operating system being used on board a ship. By contrast, the VxWorks executable image was (and still is) approximately 900 kilobytes, and the manufacturer offers complete visibility into the VxWorks source code.

Since the group was at the forefront of COTS implementation for Aegis, the unique situation arose where configuration control of the compiler and linker was not initially performed under the auspices of Aegis system CM. During the period of time where the group "owned" the compiler/linker, the ability to reproduce the configuration currently installed on the development systems (given various patches and version changes) was lost. This situation has since been rectified, in that Aegis CM has assumed configuration control of all COTS development and operating environment (compiler, linker, and support tools) components. All such purchases, installations, and upgrades will be performed using defined Aegis CM processes.

Target Versus RTOS Versus Language

As shown in Figure 1, selection of the target processor, language (including runtime support and available tools), and RTOS are highly dependent upon one another. None of these components could be selected without significant consideration for its impact to the others. For instance, it would have made no sense to choose a target for which a viable RTOS was not available. It was the successful selection of the right combination of these three components that was key to the success of the ACTS

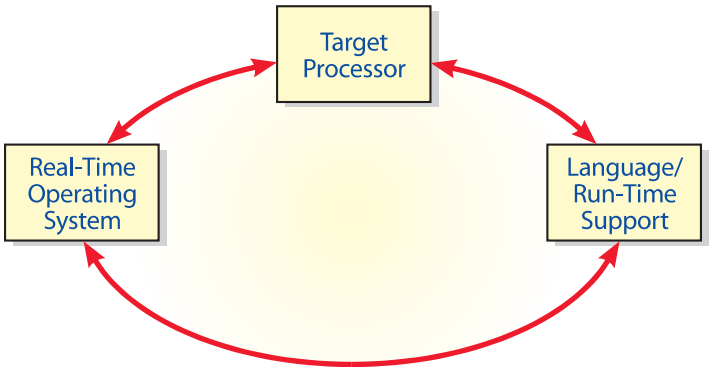

Figure 1—Combination of the Three Components

Mk 50 rehost effort. This effort is described in the following several paragraphs.

Single-Board Computers

The Training Program Development Group started the single-board computer (SBC) study with 71 board manufacturers. These manufacturers used over 25 different onboard processors between them. Required features of the board's topology and throughput (central processing unit (CPU) type and CPU speed, onboard memory, VME transfer size, VSB support, and VME interface chip manufacturer) quickly narrowed the choice to three manufacturers. At this point, the group had to simultaneously consider the system requirements, target operating system, programming language, and basic system architecture. Given the lone high-level system architecture option, VMEbus attributes were important selection criteria for the candidate SBCs. The group defined core capabilities of an SBC built for a VMEbus system. One of the fundamental strengths of the VMEbus system is also its weakness—there is, by default, only one communication path between the VMEbus components. The group recognized the need for additional communication paths between processors and chose to include a VSB. When the ACTS Mk 50 architecture was refined with alternate communication paths and known throughput requirements, the performance requirements for each characteristic of the SBC became clear. Given the I/O-bound nature of the system, SBC selection was not based solely on CPU performance. The group chose the Motorola MVME-166 SBC for its local bus architecture, VME interface chip performance, VSB support, product stability, mainstream commercial use, and RTOS and Ada compiler target availability.

Posix-Compliant, Real-Time Operating System

The group selected VxWorks as the RTOS for the ACTS Mk 50 project. No program errors have been attributed to the RTOS in the more than 4 years that it has been used, and no product patches have been required during that time. The main attributes resulting in the selection of VxWorks—and subsequent success of that selection—were:

- portability
- stability
- applicable standards compliance
- performance
- scalability
- footprint

External Interfaces

The Training Program Development Group assigned network interfaces to the SBC interface chips where possible (i.e., SCSI and ethernet). No technology was available in 1994 to implement the BFTT FDDI interface via the SBC. The architecture team selected the Interphase 5211 FDDI card, though not based on the speed of the onboard processor or advertised "throughput rate." The selection of this card was based on the availability of commercial drivers and the ability of those drivers to fit into the overall ACTS architecture, which became more refined as SBC and RTOS selections solidified.

The group considered NTDS interface boards from two manufacturers. As with the SBC and the FDDI interface cards, selection was not based exclusively on the board's speed. Both ACTS Mk 29 and Mk 50 use less than 20% of the maximum throughput on any NTDS interface because of message speed limitations that originate from the overhead associated with the NTDS protocol. There were three important factors in the selection of Visicom for the NTDS interfaces in ACTS Mk 50. First, Visicom boards have an instruction set that closely mirrors the UYK-43 I/O controller set, allowing for reuse of NSWCDD expertise in implementing drivers. Second, the Visicom boards support concurrent chains, lowering the risk of timing errors in intercomputer protocol. Third, the Visicom boards provide better VMEbus performance, which was considered more important than NTDS interface performance, since slow accesses of VMEbus by the NTDS board can cause idle time in the SBCs.

The Training Program Development Group chose magneto-optical disk drives for data recording and

scenario/lesson peripherals based almost entirely upon the desire for commonality with the proposed AWS Baseline 6 Phase I next-generation peripheral data recording device.

Defining the Architecture

For the purposes of the ACTS Mk 50 project, the task of developing the system architecture entailed the arrangement of the hardware items in a systematic manner and the subsequent allocation of training requirements to specific processors. Architecture definition was performed concurrently with component selection. It was obvious that the system components could not be selected until an architecture framework was defined. Conversely, the system architecture had to be adapted to exploit the strengths of components that were selected to meet the system design objectives. In this way, architecture refinement and component selection proceeded in parallel.

ACTS Mk 29 resides in an AN/UYK hardware architecture that has multiple (7) busses moving data through a global memory system. The ACTS Mk 29 AN/UYK CPU was observed to be over 95% busy, indicating good instruction pipeline, bus snooping, and local bus design—the CPU is rarely, if ever, waiting for a bus action to fetch instructions or data. These design attributes were considered important for ACTS Mk 50 to capture. However, the ACTS Mk 29 architecture had an undesirable attribute for a real-time, object-oriented design—90% of ACTS Mk 29 data was global. The large amount of global data was responsible for high maintenance costs for ACTS Mk 29 because unexpected areas of the program were often impacted when global data was changed. ACTS Mk 29 had a historic distribution of data and instructions to memory modules, and a distribution of tasks to processors and associated priorities of tasks within that scheme.

Had the resources been available, ACTS applications would have been completely redesigned during the rehost effort. However, only 24 months were available to perform all the tasks required to deliver the computer programs to the fleet. It was necessary to gain a full understanding of the ACTS Mk 29 design and attempt to adapt the design in the most efficacious manner possible for use in ACTS Mk 50. Certain features of the system were too high risk to alter in a 2-year period.

A limited amount of data encapsulation was implemented to reduce the system's reliance on trouble-prone global data. The ACTS Mk 50 architecture was developed to maximize throughput and performance of the CPU. The Training Program Development Group determined through analysis that, because a CPU in a COTS SBC does not have the intricate local bus features (such as the number of busses) of the AN/UYK-43, the wait states of the CPU would be more frequent. It was determined that the COTS CPU would be fully utilized at a maximum of about 75% busy in a realistic I/O scenario. Three distinct global busses (VME, VSB, and ethernet) were implemented in the ACTS Mk 50 system to allow multiple transmission paths for interprocessor messages. The theory was that, with multiple data transmission paths, the bus and CPU loads could be balanced, so there would be no single bottleneck in the system. The group took the time to understand the three global busses and their relationship with local busses on all components of the ACTS system. The group defined objectives for each of the busses and each of the three SBCs. Specific board-level design decisions were then made to meet the architecture objectives. (These decisions included the bus request level of each SBC, when DMA transfers will be used, etc.) For each piece of data passed between two or more components of the ACTS system, the group gained a complete appreciation for the data access and delivery costs, granularity, and reliability.

Verifying the Architecture

According to Software Engineering Institute's Software Capability Maturity Model, computer system architecture verification is best performed by an independent group of computer architects. In 1994, there was no such group available to the Training Program Development Group. To fill the void, detailed measurement and assessment were

done on the proposed architecture to validate all assumptions on performance of each component.

One system architecture failure was allowed for in the original ACTS Mk 50 development schedule. Fortunately, the group's initial design proved to be practicable. However, the group recognized, and felt it was necessary to mitigate, the significant risk of invalid architecture solutions by allotting adequate time to recover from an architecture failure. It follows that a particular architecture must be recognized as a failure before application code is written for it. In order for the architecture to be identified as a success or failure, criteria must be written containing discriminators that separate usable architectures from those that are unusable.

It is significant that the Training Program Development Group did not use the common method of relying on industry benchmarks to determine ACTS Mk 50 architecture viability. Industry benchmarks are designed to provide a statistical value used to compare a specific performance parameter of different systems. The group believed this method had a serious weakness—the statistical value represents an average of a series of abstractions of system performance derived under contrived circumstances. Instead, the group measured the lower, atomic levels of performance constituting the high-level benchmark. This information was then used to build a more detailed profile of how the system components executed in concert with each other. With this understanding of how the system performed and how long each step of that performance took, a much clearer understanding of system architecture was obtained than deriving a single statistical value.

An example of the Training Program Development Group's architecture verification approach is shown in the analysis of VMEbus performance. A common benchmark would have been to count the number of 200-byte messages that could be sent between two SBCs in a specified period of time. The obvious weakness of this method is that the system is doing nothing but sending VMEbus messages. In this scenario, there is no local bus, VMEbus, or memory contention; and it is likely that the entire benchmark program is running from cache memory. The resultant system profile is not representative of an actual operating environment. Instead, the group measured the atomic components of the VMEbus transfer:

- Local bus access time
- CPU instruction execution
- VME interface chip performance (arbitration time, data transfer time, and bus release time)
- Memory access time

Additionally, variables such as cache size and memory location were used to further elucidate the system profile. Once these atomic performance numbers were compiled and the execution profile of the system was determined, the group was able to determine the performance of any industry benchmark whose profile was known before it was executed on the ACTS system. The group could also define specific flaws in a product that would not make it useful for I/O-intensive, real-time systems.

Application Implementation

The Training Program Development Group selected Ada 83 for the implementation of TCP and LCP. In retrospect, it is not clear whether the group would have made that choice had the Department of Defense's Ada mandate not existed at the time (1994). However, an objective view of Ada versus C would probably have resulted in the same outcome, based on Ada's built-in concurrency, powerful typing, object-orientation, and the characteristics (such as the package specification and body) it derived having been developed specifically as a software engineering language.

Legacy Requirements

Since the legacy requirements of the system were based on an implementation in the CMS-2 language and an Aegis tactical executive system runtime, the first application development task was to map CMS-2 module entrances to Ada constructs. The Ada task construct is used to provide concurrent threads of execution in the ACTS Mk 50 system. Detailed measurement of the ACTS system and the Ada runtime environment showed that task synchronization via Ada's rendezvous

feature was time prohibitive. (The rendezvous cost is 200 μs, the same as a CMS-2 successor entrance call.) Therefore, in time-critical threads, semaphores are used for task synchronization.

As mentioned previously, limiting the amount of global data in the ACTS system via information hiding was a major goal during the rehost. As processes were reallocated to Ada, data was assigned to specific tasks, where possible. The design of ACTS Mk 29 was such that, without an extensive redesign, most ACTS data would still need to be shared by multiple tasks. Where a global data scheme was retained, a single task was designed as a data controller, and data access was achieved through procedure calls to the controller task.

Distributed Interactive Simulation (DIS) Interface

One consequence of acquisition reform is the move away from government and military standards. The Training Program Development Group was required to implement an ACTS Mk 50 interface subject to the DIS standard to communicate with BFTT, the LAUs, and external entities (such as Theater Missile Defense System Exerciser). DIS is an academic-driven standard to which changes are made without regard for fielded systems. The implications of using a fluid standard in a controlled shipboard environment complicated change management of ACTS Mk 50. Like the VMEbus standard (IEEE-1014), the DIS standard is open to interpretation, and interoperability is not guaranteed by adherence. During development, ACTS Mk 50 (along with BFTT) was tested in several types of joint exercises (e.g., Kernel Blitz, STOW-E (Europe), and Theater Missile Defense System Exerciser) led by various program offices. It was observed during these events that no two exercises used the same implementation of IEEE-1278. In general, implementation of any COTS standard was found to remove responsibility for, and control of, the interface protocols from the development group and introduce previously unencountered risk factors.

Code Reuse

The Training Program Development Group designed and implemented postmission analysis reusing ADAR system components. The only new code required provides an RD-358 (9-track) tape interface and an operator shell around the ADAR applications. The ADAR code reuse was possible largely because the Support Software Group at NSWCDD had the foresight to create platform independent, portable ADAR libraries. The reuse approach saved the cost of developing data extraction and report generation tools, and printer drivers. The savings from the reuse are even more significant since the problems associated with the variability of recorded data across Aegis program baselines are handled by ADAR via data dictionaries. Also, PMA can implement ADAR code upgrades made by the Support Software Group, avoiding the need for code maintenance for PMA. Therefore, the ADAR code reuse provides recurring cost savings throughout LSE.

The Training Program Development Group built the LCP using Ada code from a program owned by the ATRC that hosted computer-aided submode training (CAST) within console emulators. The Ada code provided by ATRC was modified by the Training Program Development Group to fit the specific ACTS architecture and to support multiple simultaneous CAST lessons. Unlike PMA, the LCP reuse was a one-time saving because the Training Program Development Group performs LCP code maintenance.

While more often a worthy concept than a workable practice, it is notable that ACTS Mk 50 has these two significant and successful instances of code reuse.

Superset Philosophy

From the start of the project, the Training Program Development Group designed the ACTS Mk 50 applications with a superset philosophy. Two basic methods were used to meet this goal. In some cases, simple processing variations for different Aegis baselines were handled using control logic in the program. In other cases, modules dealing with major functions that differ by baseline were coded to be interchangeable at build time. The core functionality of the system remained in a set of packages that are applicable to all program baselines, thus greatly simplifying the maintenance task.

By adopting the superset philosophy, the group once again found itself at the forefront of a relatively untried practice. As a result, the Aegis configuration status accounting system was somewhat confused by the appearance of program changes that were not approved for a defined upgrade. This occurred because all packages, regardless of their applicability to the baseline, were delivered as a library to the Configuration Management group. Baseline-specific build files were used to select the subset of Ada packages to be compiled and linked to produce a specific configuration and/or program baseline. As this process has continued, progress has been made in addressing these concerns. The point of this particular anecdote is that other groups attempting to implement this philosophy may encounter the same initial difficulties.

Ship Delivery

NSWCDD delivery of a COTS-based component of the AWS to ships where none previously existed presented new crew training, LSE, and maintenance challenges. The effort to document the system for third-party assembly and to develop crew training packages for ACTS Mk 50 was significant.

Third-Party Vendor System Assembly

AP Labs in San Diego, California, was selected to build production ACTS Mk 50 systems as part of the BFTT system assembly. Having engineered the system, the Training Program Development Group was required to create a system assembly document to describe the requisite components, their placement, all jumper and switch settings, every fuse and battery that requires verification, and boot parameter definitions for the SBCs. The group also created a Critical Item Development Specification to document requirements for the VME enclosure, internal and external connections, and external indicator lights. It seems reasonable to assert that comparable documents must be created for any ship-delivered COTS-based system.

AP Labs was tasked to procure the VME chassis, power supply, and rack in accordance with data and specifications provided by the Training Program Development Group. The Training Program Development Group created a set of diagnostic programs to verify that all switches and jumpers in the system are configured properly. Once the system is assembled, the diagnostic tool is used to verify the correct assembly of the system. For the three ship installations performed thus far, the process has worked extremely well.

Training Package Generation

Since the ACTS system was an aggregate of many COTS components, an overall training package did not exist for the system. Each component of ACTS Mk 50 was delivered to the assembler with a vendor-supplied technical manual, but it was not realistic to expect a ship's crew to isolate a system problem with these documents. For the ACTS Mk 50 system, there were over 16 technical specifications required to fully describe the system. (The resulting "document stack" is over 2-ft high.) The Training Program Development Group extracted key facts from the appropriate technical manuals, added significant facts for maintenance of a ship-based system, organized the data in a manner conducive to problem solving, and created a training and reference package for the ACTS Mk 50 System. Creation of this type of training document is a new requirement with the introduction of COTS systems. Previously, training material for the computing systems such as the AN/UYK-43 was provided in other existing documents. The detail of the information required by a ship's crew was a source of debate. This is an issue that will continue to be addressed, as new COTS systems are installed aboard ships for some time to come.

Shipboard Installation and Checkout

The NSWCDD Aegis Program Office rarely installs a complete system (i.e., equipment and computer programs) aboard a ship. Since the Training Program Development Group defined the ACTS Mk 50 system, it was incumbent upon that group to define procedures to verify the equipment installation before installing the software. A process was defined to validate every type of board-to-board transfer of data using ACTS Mk 50 diagnostic programs. During the shipboard checkout, the following items are verified:

- Hardware settings
- Boot EPROM settings
- Battery power of all SBC batteries
- IP addresses of all ACTS nodes
- Serial numbers of all hardware components
- Firmware revisions of all hardware components that historically have frequent change

System Maintenance

As a software-oriented entity, the Training Program Development Group found itself in a number of previously uncharted waters during ACTS Mk 50 development. The derived requirement for intimate hardware familiarity associated with creating the system architecture and performing component selection has been discussed. In addition, the group was required to perform hardware maintenance and troubleshooting for the rehosted system at all sites and onboard ships. Those typically responsible for maintaining hardware were not yet ready to assume that role for COTS-based systems. While the intent exists for a more traditional separation of hardware and software maintenance to eventually be restored, there is little doubt that, in the COTS arena, software development groups will be required to accept much more responsibility for the systems' hardware than was previously necessary.

REFRESHING THE TECHNOLOGY

One of the theoretical COTS technology benefits is the capability to easily implement desirable technology advancements. Clearly, any implementation that is not designed with a set of alternate growth paths is not exploiting this significant potential of COTS technology. The benefit of COTS technology advancement, which can also be viewed as planned obsolescence, also provides a new challenge. Implementers of COTS-based systems must plan for 3- to 5-year technology upgrades; otherwise, large, up-front product purchases will be required. The main technical problem with the latter option is the unavoidable lack of vendor support for commercial products as they age.

ACTS Mk 50 was created with mainstream, standards-compliant components that provide multiple paths for future growth. The benefit of assuring portability in the architecture definition allows for relatively painless technology upgrades. In Aegis, the upgrade of commercial technology is referred to as a "technology refresh." There are two ACTS Mk 50 refresh efforts underway. A major technology refresh is being pursued to coincide with the Aegis Baseline 5.3.8 effort. This refresh will include an upgrade to the PowerPC MVME-2700, Ada 95, and VxWorks 5.3. In each case, a fairly straightforward move within the product's growth path is all that is required. This is not meant to understate the effort, which will still be significant, but it will carry much less risk than many of the original rehost alternatives would have. In addition to the major refresh effort, a smaller effort is being performed to update ACTS Mk 50 to an MVME-167 SBC. This is required because of the discontinuation of the MVME-166 product line and the required schedule for supporting deploying Aegis ships. This SBC upgrade could have been catastrophic had not an SBC been selected for which multiple, mainstream upgrade paths were available. The importance of using proven COTS mainstream products cannot be overstated.

Although ACTS Mk 50 has been in the fleet since February 1998, the Aegis LSE program, as a whole, is still fairly immature with regard to COTS technology. As the Training Program Development Group begins its first refresh effort, an independent architecture verification group has yet to be established for Aegis. Responsibility for verification of the technology refresh architectures remains with the Training Program Development Group.

With the upgrade path of ACTS Mk 50 computing being a simple continuation of the proven technology path commenced in 1994, the technology refresh will be performed within an AWS maintenance baseline. The ability to perform a COTS technology refresh within the time constraints of a normal maintenance cycle is a goal worthy of any project. ACTS Mk 50 has achieved that goal.

BENEFITS

The rehosting an entire element of the AWS at NSWCDD provided corporate, training and cost benefits. The corporate benefits include a great expansion of the corporate knowledge base; creation of a detailed understanding of the new paradigm in the life-cycle model by creating an entire product; and hands-on knowledge of COTS hardware, software, and architecture evaluations. This will provide necessary experience when NSWCDD is tasked to perform LSE of COTS-based Aegis baselines.

Historically, Aegis training capabilities have been limited by the target computing platform. With the rehosting of the ACTS computer programs to a mainstream system and the underlying ACTS computing model being subtly changed to support the new model, the ACTS system is poised to accept technology improvements that will continue in the commercial sector. In the 17 years between 1981 and 1998, ACTS Mk 29 computing capabilities changed once (AN/UYK-7 to AN/UYK-43), with an approximately twofold processing power increase. With the program now hosted in a COTS-based system, it is estimated that ACTS computing power will triple with each refresh effort, with refresh efforts occurring every five years. Therefore, by 2010 (after three refresh efforts), ACTS will have 15 to 20 times the computing power it would have had on a historic military computing upgrade path. With the increased capabilities, ACTS can manage more tracks and provide more interfaces with other COTS systems. These interfaces will allow ACTS to better participate in joint training and will provide for more effective battle group training.

CONCLUSION

The development costs of a new standard Navy computer are high. The engineering costs of COTS products are also high when considering the full life-cycle cost, technology refresh, specifying requirements for COTS-based systems, managing COTS products changes while maintaining a safe weapons system, and providing proper system maintenance and training. However, the processing capability benefits of a successful technology rehost can be enormous. The efforts undertaken and successfully accomplished during the ACTS Mk 50 rehost effort led to a number of lessons learned and suggestions that could be helpful to any other group that is considering or performing such an effort.

ACKNOWLEDGMENTS

Neil Lauver and Warren Jones of Computer Sciences Corporation, King George Office, performed the studies that resulted in component selection for ACTS Mk 50, and they defined and validated the ACTS Mk 50 system architecture. Most of the technical information in this article can be credited to the work of these gentlemen.

BIBLIOGRAPHY

Abowd, Gregory et al., *Recommended Best Industrial Practice for Software Architecture Evaluation*, Software Engineering Institute Technical Report CMU/SEI-96-TR-025.

ACTS Rehost CMS-2 to Ada Language Mapping, Computer Sciences Corporation, Integrated Systems Division, King George Office, 26 Jul 1994.

ACTS VME-Based HWCIs Technical Report, Computer Sciences Corporation, Integrated Systems Division, King George Office, 5 Apr 1994.

ACTS Rehost Programmer's Manual, Computer Sciences Corporation, Integrated Systems Division, King George Office, 19 Sep 1995.

ACTS Rehost VME Hardware Architecture Computer Sciences Corporation, Integrated Systems Division, King George Office, 1 Sep 1995.

Lauver, Neil and Jones, Warren, *Aegis Expanding Training: Enabling Team, Fleet, and Joint Simulations*, Computer Sciences Corporation, Integrated Systems Division, King George Office Sep 1997.

THE AUTHORS

MR. GREGORY D. MILLER

Mr. Gregory D. Miller was a computer scientist in the Training and Simulation Branch of the Combat Systems Department. He graduated from the University of Pittsburgh at Johnstown in 1982 with a degree in computer science. He has been employed at NSWCDD since April 1983, when he joined the Aegis Combat Training System Group. Mr. Miller developed computer program changes for shipboard applications, defined high-level requirements for the ACTS Scenario Development System, implemented a shore-based operator training system for the Aegis Training Center, and led the ACTS Mk 50 development team. Mr. Miller is currently an engineer on the CVX Program in the Advanced Control Systems Engineering Branch.

MR. JEFFREY LIPE

Mr. Jeffrey Lipe has a degree in computer science from Park College. He has been associated with the Aegis program for over 13 years in the areas of configuration management, quality assurance, and software development. He is currently a software engineer for the Training Program Development Group in the Training and Simulation Branch of the Combat Systems Department.